Lehigh University
Lehigh Preserve

Theses and Dissertations

1-1-1983

The enhancement of IBM solid to include geometric constructions in a plane.

Carol Marie Riggin

THE ENHANCEMENT OF IBM SOLID
TO INCLUDE GEOMETRIC CONSTRUCTIONS IN A PLANE

by

Carol Marie Riggin

A Thesis

Presented to the Graduate Committee

of Lehigh University

in Candidacy for the Degree of

Master of Science

in

Industrial Engineering

Lehigh University

1983

CERTIFICATE OF APPROVAL

This thesis is accepted and approved in partial fullfillment of the requirements for the degree of Master of Science.

5/11/83

(date)

Professor in Charge

Chairman of Department

ACKNOWLEDGEMENTS

I wish to thank Bill Fitzgerald, Franklin Gracer and Bob Wolfe of IBM for their continual support and coaching. I also wish to thank Dr. Louis Plebani, my major advisor, for providing technical guidance and Dr. Emory Zimmers, my minor advisor, for obtaining the resources necessary to complete this project.

This thesis is dedicated to my parents and to my husband, Glenn, for their constant support throughout the years.

Table of Contents

List of Figures

List of Tables

ABSTRACT

The IBM SOLID system models complex 3D objects by combining volume primitives. These primitives are geometric volumes such as cuboids, cylinders, and swept polygons. IBM SOLID uses an interactive graphic input subsystem called GRIN (GRaphic INput), which aids the mechanical designer in easily entering and manipulating primitives. GRIN allows the user to translate, rotate, and scale the primitives, as well as to combine them using the set operations of union, difference, and intersection.[1] With the implementation of user interfaces such as GRIN, solids modeling becomes a viable and promising alternative to traditional design.

The purpose of this thesis is to expand GRIN's capabilities to include geometric constructions in a plane. This requires the user to first establish a working plane by specifying its orientation in 3D space. (The user will in effect be defining a true local coordinate system). Points, lines, circular arcs, and circles can then be created by using such geometric relations as intersections, tangents, and normals. The

---

[1] W. Fitzgerald, F. Gracer, R. Wolfe, "GRIN: Interactive Graphics for Modeling Solids," IBM J. Res. Develop. 25, 1981, pp. 281-294.

results can be used as construction aids for defining 3D objects, especially to create polygons for translation or rotation.

This thesis involves the development of an interactive system to create geometric constructions in a plane. A top down structured approach was used in developing the design of the system. Special emphasis was placed on the user interface to the system, especially in the areas of menu design and response time. The detailed design of the system is included and can function as an operator's manual.

1. BACKGROUND

Solid modeling systems enable users to create models of 3D objects out of solid shapes. In development since the late 60's these systems are now emerging in the marketplace in full force.[2]

According to Kinnucan[3], there are two main reasons for the recent emergence of these systems. First, the computational costs required for solid modeling have been drastically reduced due to technological advancements with computer processors. Previously, computer costs were a significant market restriction.

Second, users are becoming more aware of the capabilities of Computer Aided Design/Computer Aided Manufacturing (CAD/CAM) systems. Users have witnessed rapid technological growth in these system capabilities from 2D drafting aids to 3D wireframes. In addition non-drafting tasks have been added such a producing and verifying machine tool programs and performing 2D finite element analysis.

Solid modelers provide significant advantages over

---

2 P. Kinnucan, "Solid Modelers Make the Scene", High Technology, July/August, 1982, p. 38.

3 IBID., p. 38.

current 3D wireframe systems. According to Kinnucan[4] and to Requicha and Voelcker,[5] the capabilities of solid modelers are many and include:

1. Wireframe representations are ambiguous, in that a wireframe drawing could have several interpretations. With solids, hidden lines can be automatically removed to clarify the drawing. Shading can also be added to produce an even more realistic drawing.

2. "Nonsense" objects are tolerated in wireframe systems. The user, rather than the system, must detect such anomalies as a missing edge. Solid modelers avoid this problem.

3. Interference checking can be performed with solid modelers to determine whether objects are touching one another. This aids in the kinematic analysis of functional parts (as in robot simulations) as clearances between parts can be continually checked. This reduces the need for mechanical prototypes.

4. Users of wireframe systems must supply a lot of low-level data to describe a part. Solid modelers use powerful model building operations. Designers can create complex models quickly from simple solid shapes.

5. Solid modelers allow for automatic process planning and machine tool verification. The models on current design systems do not provide sufficient information for these

---

[4] Kinnucan, op. cit., pp. 39-40.

[5] A.A.G. Requicha and H.B. Voelcker, "Solid Modeling: A Historical Summary and Contemporary Assessment", IEEE Graphics, March, 1982, p. 10.

functions.

6. Solid modelers enable finite element analysis to be performed automatically on 3-D objects. The results of the analysis can be displayed to enable engineers to visualize the effects of such factors as stress and heat on a part.

7. Solid modelers can be used to define objects to a robot thus; robots can recognize objects in any spatial orientation.

There are two main approaches to defining solids in the computer: constructive solid geometry (CSG) and boundary representation (B-rep). In CSG, the object is described in terms of elementary shapes, or primitives (ie. cuboids, cylinders and hemispheres). These primitives are linked together in a tree structure with the Boolean operations of union, difference and intersection.[6]

In B-rep systems, a solid is represented by its boundaries. The model is stored as a linked list of faces, edges and vertices. B-rep systems allow the user to create swept solids by translating or rotating a 2D section of an object. With a B-rep system, a user must typically draw all of the edges of the solid and indicate

---

[6] R. Hillyard, "The Build Group of Solid Modelers", IEEE Graphics, March, 1982, p. 44.

the relationships among them.[7]

This thesis addresses IBM's solid modeling system, "IBM SOLID". This system models complex mechanical objects from volume primitives. Within IBM SOLID, a model is represented as a hierarchical structure which retains primitives at the lowest level.[8] IBM SOLID is mainly a B-rep system but the capability of CSG also exists. The system largely uses polyhedral representations in which objects are defined with polygon faces.

IBM SOLID uses an interactive graphic input subsystem called GRIN (GRaphic INput), which aids the mechanical designer in creating and manipulating objects. Although IBM SOLID is a B-rep system, GRIN allows the user to input volume primitives (ie. The user can define a cuboid by entering 4 points). In addition, GRIN allows the user to translate, rotate or scale solid objects, as well as to combine them with Boolean operations.[9]

----

[7] Kinnucan, op. cit., p. 40.

[8] W. Fitzgerald, F. Gracer, R Wolfe, op. cit., p. 281.

[9] IBID., p. 282.

It is the purpose of this thesis to expand GRIN's capabilites to include geometric constructions in a plane.

2. PROBLEM DEFINITION

The IBM SOLID system is limited in its ability to generate polygons which can be translated or rotated into solids. Currently, polygons can only be created by connecting lines and arcs in a series, with lines being defined only by entering two end points, and arcs only by entering two surface points and a center point. The need existed for the mechanical designer to create 2D elements (points, lines, arcs and circles) using construction aids. For example, users desired the ability to create a circle by selecting three tangent lines.

In addition, construction aids were needed to define positions for the creation and placement of volume primitives. The 2D elements mentioned above can be used to define these positions for the volume primitives. For example, a point created using construction aids could be selected as a corner point of a cuboid. Thus; 2D construction aids would be an excellent productivity enhancement to the current solids system.

The specific tasks of this thesis involve defining and implementing the capabilities for the GRIN user to:

- Create 2D elements on a working plane using geometric constructions.

  For example, the user can create a circle by selecting a center point and a line to which the circle will be tangent.

- Edit the 2D elements within a working plane. This includes rotating, translating, copying, scaling, refaceting, erasing, and segmenting these elements.

- Generate volume primitives using 2D elements from a working plane. For example, points created on a working plane can be used to define a cuboid or, a polygon in a working plane can be translated to create a solid.

- Create 2D elements using solid objects. For example, a point can be created in the working plane by projecting a vertex from a solid on to that plane.

In accomplishing the above tasks, the issue of user friendliness was a big concern. For productivity gains to be realized, the construction aids must be designed for ease of use by a mechanical designer.

3. METHODOLOGY

A top down modular approach was used in the development of this system. Major design decisions were made at the top levels before much time had been invested in defining (or coding) the details. The various functions of the system were addressed in a modular fashion, thus facilitating a more understandable and maintainable system.

Although many tasks overlap, this section has been divided into three phases for clarity: analysis, structured design and implementation.

1. ANALYSIS

a. Definition of System Constraints

System constraints which would limit or confine the system design were identified. Both the hardware and software environments were defined.

b. Definition of User Requirements

The basic system requirements were determined through interviews with the user group. In addition, other computer graphics and numerical control systems were investigated. These included CADAM from Lockheed, the Applicon 880 system, and APT (Automatically Programmed Tool).

System requirements were documented in a comprehensive user's guide which served as the basis for the entire sytem design.

2. STRUCTURED DESIGN.

a. Design of the Basic System Structure

The major functional areas of the system were identified. These functions were arranged hierarchically and served as the backbone for the entire design process.

b. Design of the System Outputs

The design of the system menus and messages took into account the following factors:

- Ease of Use
- Clarity of Prompts and Messages
- Experience of the User (ie. The use of defaults for the novice user)
- Error Handling
- Number of Entries Required to Execute a Function

c. Design of the Data Structures

The file structures (in particular the design for the model and display files) were designed not only to be efficient for the proposed system but also to have a minimal impact on the current system.

d. Detailed Design of the Modules

Detailed modules were designed which expanded and defined the top level functional areas. The modules were designed to take advantage of the structured design concept. Thus whenever possible, they were small, independent, "black box" modules. Each module generally represented a particular control or processing function. Pseudo-code and/or flowcharts were used to document these modules.

3. IMPLEMENTATION

a. Development of a System Test Plan

To insure successful integration into the current system a detailed test plan was developed .

b. Implementation of Coding

PLI was the programming language used for this system. Performance considerations (ie. Response time) were taken into account at this time. In addition, since many of the modules were performing common mathematical calculations, the use of currently developed software (ie. APT) was investigated.

Since the coding of this system is beyond the scope of this thesis, this task is not completed at this time.

c. Development of User's Training Guide

A comprehensive but easy to follow user's guide was written for documentation and training purposes.

4. CONSTRAINTS

The system constraints which would limit or confine the system design were identified. They include:

1. HARDWARE

   "The hardware configuration consists of a standard IBM 3270 alphanumeric display equipped with the Graphic Attachment feature, which allows a storage tube display such as a 19-inch Tektronix 618 to be attached. This results in a dual screen display station with graphic commands transmitted to the display head at a very high rate"[10] The display station is attached to an IBM 4341 processor with 4 megabytes of core memory.

2. SOFTWARE

   IBM SOLID runs under the VM/CMS operating system. PLI is the standard source language used in developing the system. Assembler routines can be called from PLI.

3. TIMETABLE

   Lehigh University currently has a three year research contract with IBM to provide enhancements to IBM SOLID. These enhancements include not only planar constructions, but also dimensioning and tolerancing.

---

10 W. Fitzgerald, F. Gracer, R. Wolfe, op. cit., p. 291.

5. SYSTEM DESIGN

This system establishes a working plane defined in 3D space. The user can specify the origin and the orientation of a Local Coordinate System (LCS) which defines the plane. (See MODEL COORDINATE SYSTEMS.) Once the working plane has been established, the user is limited to creating and editing 2D elements in that plane. Points, lines and circles can be created using construction aids such as tangents, normals and intersections. These 2D elements can then be manipulated in the plane. Elements can be erased, moved, turned, scaled, reflected, copied and relimited.

The system structure chart is shown in figure 5-1. As can be seen from the chart, the system was designed in a modular, hierarchical fashion. The main module "Enter Working Plane Mode" controls the execution of the entire system. It initializes a working plane or chooses a predefined working plane and then passes control to the next hierarchical module. The user can then "Create Working Plane Elements" or "Edit Working Plane Elements". These two functions are described in the Detailed System Design chapter. These two modules then call other modules when necessary. These include:

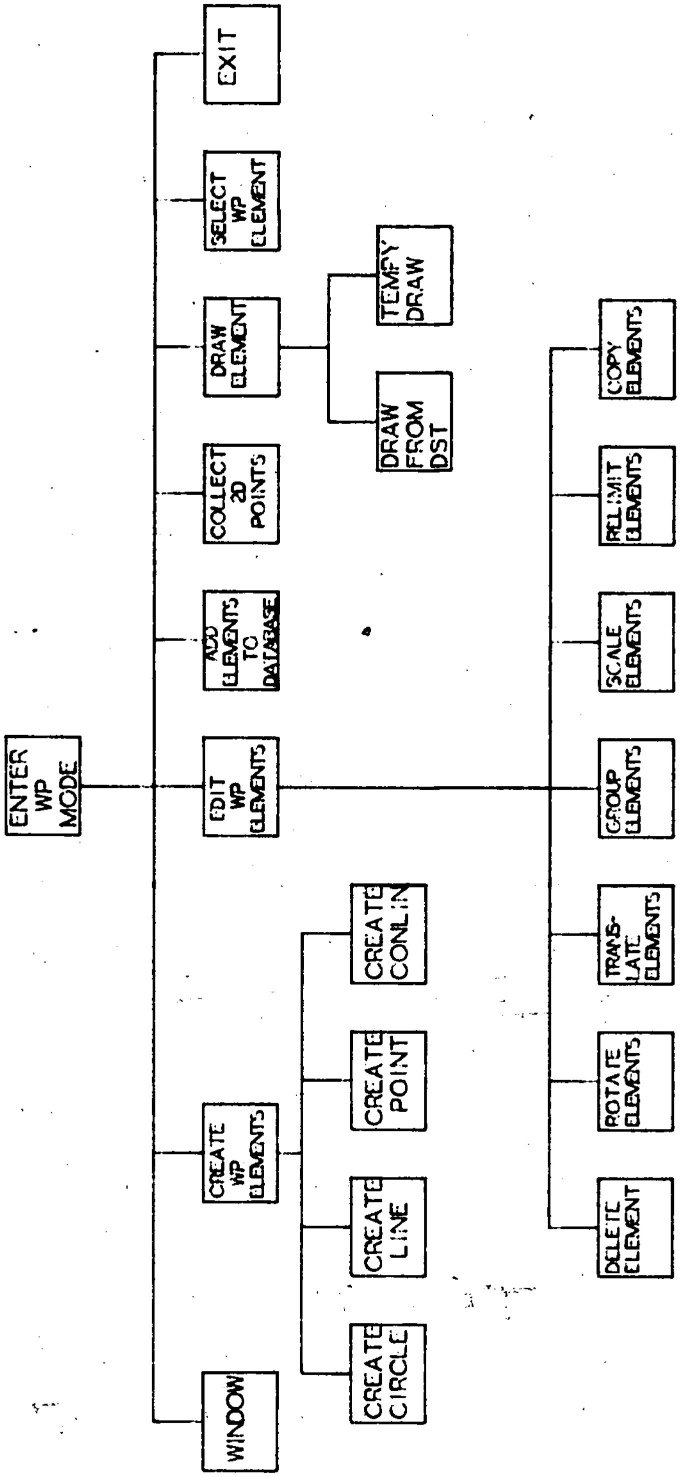


Figure 5-1: SYSTEM STRUCTURE CHART

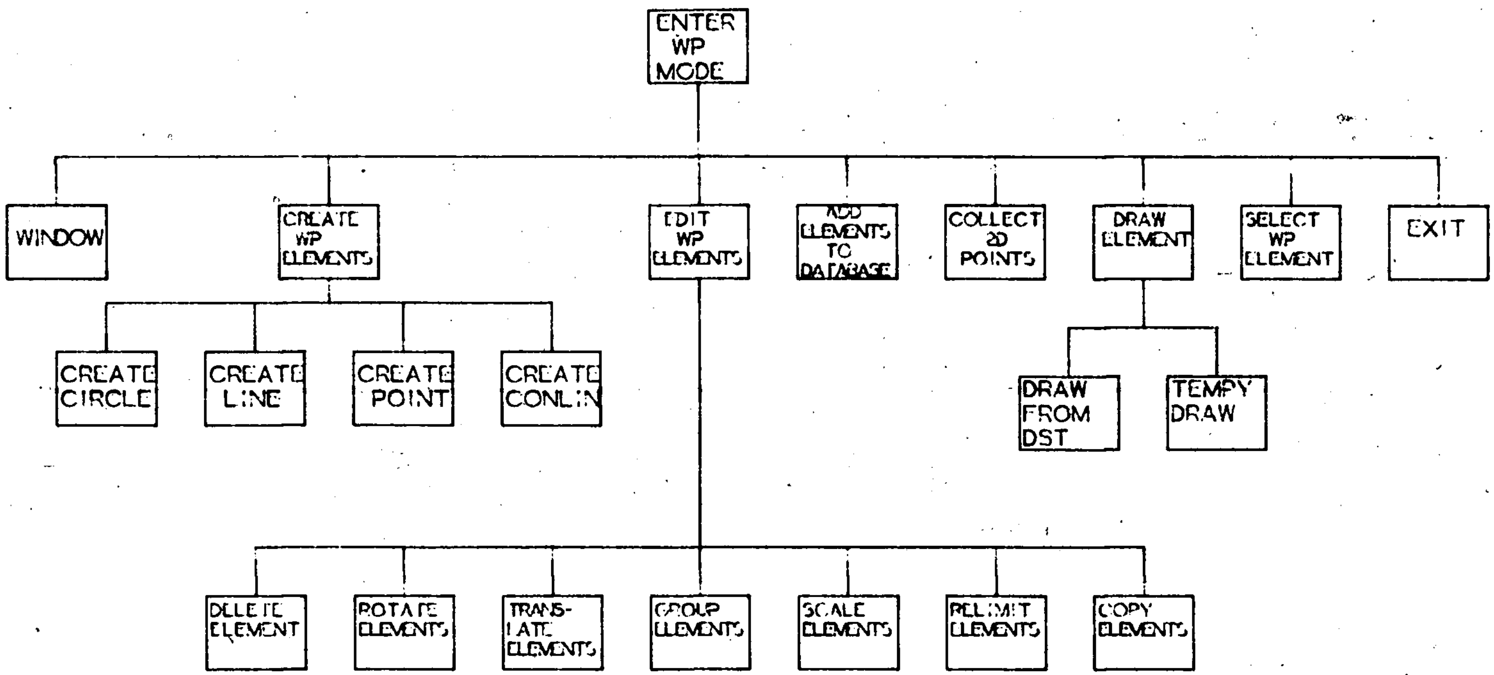


Figure 5-18 SYSTEM STRUCTURE CHART

1. Select Working Plane Elements. Given the type of element the user requires, this module searches through the display list (See Data Structure) to find the closest element to the position indicated by the user. The routine then extracts and returns the data for that element from the database.

2. Collect 2D Points. This module is called whenever a 2D position is required. This position can be defined by keying in relative or absolute u and v coordinates, indicating a vertex, or keying in a vertex name.

3. Add 2D Elements to the Database. This module adds given element data to the database. It performs the housekeeping work involved in modifying the vertex and edge lists.

4. Draw 2D Elements. When an element is first created it is displayed temporarily until the user accepts it. Once accepted, the element is placed in the display file for redisplay when necessary.

6. DETAILED SYSTEM DESIGN

A working plane is a 2D plane imbedded in 3D space. Vertices in 3D can be projected on to the plane; points, lines, arcs and circles can be created; and intersections, tangents, and normals can be constructed. The results can be used as construction aids to help in the definition of 3D objects, especially to provide polygons to be translated (PT) or rotated (PR) into solids.

The working plane is represented by a node in the database hierarchy; thus it can be manipulated by the same commands used with solids. Once the user enters the working plane however, he/she is limited to entering and manipulating 2D elements only in the specified working plane (unless specifically instructed otherwise as in the PROJ sub-command).

6.1 ESTABLISHMENT OF A WORKING PLANE

WP <N> Enter Working Plane

To create a new working plane the ENTER key must be hit immediately after the WP command. The user then has the following options:

1. to hit the ENTER key again to create a working plane at the orientation of the LCS. The working plane will have the same origin as the LCS with the +w axis being perpendicular to the working plane.

The LCS can be modified before hitting the ENTER key by using the LCS sub-command.

2. to name or select a previously established working plane. The new working plane will be oriented to this plane.

The working plane will be added to the hierarchy at the position defined by DEST and SELOBJ.

A previously defined working plane can be chosen by:

1. keying in the name <N> of the working plane.
2. selecting any element in the working plane.

Once the user has entered a working plane the following menu appears:

PT/LINE/CIRCLE/CONLIN/EDITEL/RET

The user may now create, modify or delete elements in the working plane.

6.2 CREATION OF WORKING PLANE ELEMENTS

The user can construct 2D points, lines, arcs and circles while in a working plane. These 2D elements are stored as edges of the working plane; thus there is no node associated with them.

When in the working plane, elements can be selected either by activating the graphics device ("IND") or by accepting the currently selected element. This element is the one last selected by the user. The currently selected element can be highlighted using the HIELM sub-command in many of the WP menus.

6.2.1 POINT Creation of a Point Element

A point can be created as an element in the working plane through the use of the POINT function. It should be noted that vertices are not point elements unless specifically defined as such using the POINT function. Point elements are displayed as small squares on the screen.

Upon the creation of a point, the user may reject the point, select another point sub-command or return to the main menu in working plane mode.

When creating a point, the following main menu will appear:

CP/INT/CEN/EPT/SPA/NOR/ALEN/PROJ/ANG/

A cancel command entered while creating a point will return the user to this menu.

Each of the sub-commands listed in the above menu will be described.

1. CP Current Point

   A point element is created at the current point indicator (CP). To create a point, the user must enter YN when the CP is at the desired location.

   The CP can be moved by any of the following methods:

   - LOC C1 C2 Absolute Local Coordinates

     Move the CP to the coordinates specified by C1 and C2 (u and v coordinates).

- A <N> Axial Move

  Move the CP along the A axis a distance N, where A can be either u or v. If N is not specified, the CP will be moved until its A axis coordinate is equal to that of a point to be selected.

- FN <N> Find Named Vertex

  Move the CP to the vertex whose name is N. Vertices are named using the NV (Name Vertex) command. It should be noted that a point element is also considered a vertex (the reverse is not true however). Only vertices contained in the working plane are valid for this command.

- IND Indicate a Vertex.

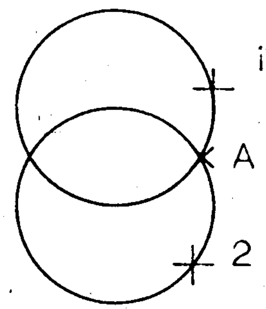
  The graphic device is used to select an existing vertex. It should be noted that a point element is also considered a vertex (the reverse is not true however). Only vertices contained in the working plane can be selected.

- C1 C2 Relative Coordinates

  Relative u and v distances are assumed if the user enters in two coordinates C1 and C2.

2. INT Intersection

A point will be created at the intersection of two elements (line, arc, or circle). Where necessary, extensions of lines and arcs will be used for intersections (ie. Lines will be considered as unbounded lines and arcs will be considered as circles). If more than one intersection point is possible, the user must select the elements nearest the desired intersection point. See figure 6-1.



The top circle was selected at 1 and the bottom circle at 2 to create point A.

Figure 6-1: INTERSECTION OF TWO CIRCLES

3. CEN Center Point of an Arc or Circle

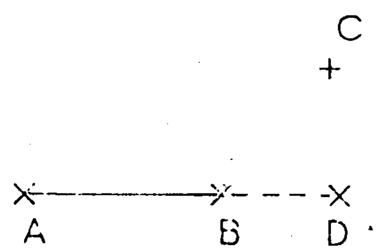
   A point will be created at the center of a selected arc or circle.

4. EPT Endpoints of a Line or Arc

   A point is defined at the endpoints of a selected line or arc. This function may be especially useful for distinguishing between contiguous collinear lines.

5. NOR Normal

   A point will be placed at the intersection of a normal line drawn from a specified point to a selected line, arc or circle. Extensions of all lines and arcs will be used. See figure 6-2.



Point C and line AB were selected to generate point D.

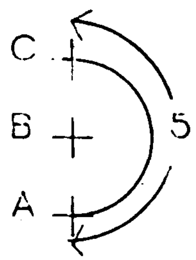
Figure 6-2: POINT NORMAL TO A LINE

6. SPACE <N> N Points Spaced Evenly

   N points will be spaced evenly along a selected line, arc or circle or between two vertices. Points are not created at the endpoints of a line or arc. For example, to create a point at the midpoint of a line or at the bisector of two points use "SPACE 1". If a circle is selected, N must be at least 2 and the user will be prompted to enter a reference point on the circle as the first point.

7. ARCLEN Arc Length

   A point will be created a circular arc length from a specified point. The center point of the arc must also be entered. See figure 6-3. All arcs will be determined in a counter-clockwise direction.



Points A and B were selected to define the start point and center point of the arc.
An arc length of 5 was keyed in to then generate point C.

Figure 6-3: POINT CREATED AT CIRCULAR ARC LENGTH

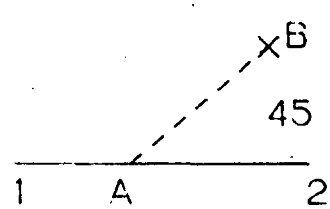
After the point has been created, the user has the option to create the point in the opposite angle.

8. PROJ Projection to Working Plane

   A point will be created by the normal projection of a specified point on to the working plane. In this sub-function the user is not limited to the working plane when specifying the point to be projected. All of the methods in the section, "Specifying a Point in 3D" can be used.

9. ANGLE Angle From a Line

   A point will be created a specified distance and angle from a selected line. A reference point on the line (or projected to the line) must be selected. See figure 6-4.



Line 12 and point A were selected. An angle of 45 degrees and a distance were keyed in to generate point B.

Figure 6-4: POINT CREATED AT AN ANGLE

After the point has been created, the user has the option to create the point in the opposite angle.

6.2.2 LINE Creation of a Line Element

A line can be created as an element in the working plane through the use of the LINE function. Where necessary, extensions of construction lines and arcs will be used (ie. Lines will be considered as unbounded lines, and arcs will be considered as circles). Upon the creation of a line, the user may reject the line, select another line subcommand, or return to the main menu in working plane mode.

When creating a line the following main menu will appear:

/LENGTH/UNLIM//POINT/TANG/PAR/PERP/ANGLE/

A cancel command entered while creating a line will return the user to the above menu.

The u or v axes may be specified whenever a line is required in any of the above sub-commands. For example, a line may be created through a point and parallel to the v axis.

Each of the sub-commands listed in the above menu will be described.

1. LENGTH/UNLIM

   The user can create lines that are either unbounded or a given length. If a natural boundary exists for a line, the line length will default to the determined length. For example, a line created between two points will be bounded by those two points. If no natural boundary exists (ie. A line through a point and parallel to another line) an unbounded line will be the default. If a line length different from the default is desired, the user must select either LENGTH or UNLIM before selecting any other sub-command. If the length sub-command is selected the system will prompt for the starting point, a length and a direction, or the starting point and a limiting element, after the line has been defined.

2. POINT

   The user must specify a point (point 1). The line to be created will be through this point.

   The following sub-menu will appear:

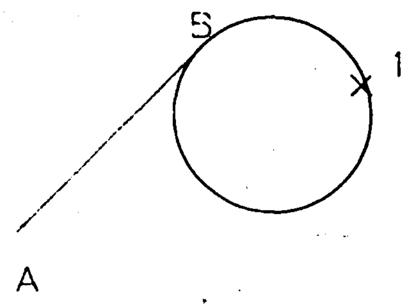
   /POINT/TANG/PAR/PERP/ANGLE

   a. If POINT is again selected, the user is prompted to specify another point (point 2). A line is created between point 1 and point 2.

   b. If TANG is selected, the user is prompted to select an arc or circle. A line is created through point 1 and tangent to the selected arc or circle. The location where the user selects the tangent arc or circle will be used to determine the desired line. See figure 6-5.

   c. If PAR is chosen, the user is prompted to select a line. An unbounded line is created through point 1 and parallel to the selected line.

   d. If PERP is chosen, the user is prompted to select any line, arc or circle. A line is created through point 1 and perpendicular to the selected element. The location where the user selects the perpendicular arc or circle will be used to determine the desired line.



The circle was selected at 1 to generate line AB.

Figure 6-5: LINE TANGENT TO A CIRCLE

e. If ANGLE is chosen, the user must select a line and key in the number of degrees the angle forms with the line. An unbounded line is created through point 1. All angles are measured in a counterclockwise direction.

3. TANG

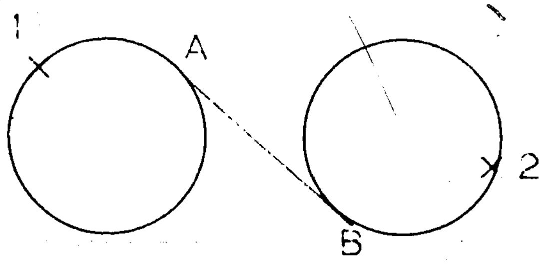
The user must select an arc or circle (elem1). The new line will be tangent to elem1. The location where the user selects the tangent arc or circle will be used to determine the desired line.

The following sub-menu will appear:

/POINT/TANG/PAR/PER/ANGLE/

a. If POINT is selected, the user must specify a point. A line is created tangent to elem1 through this point.

b. If TANG is again selected, the user must select another arc or circle (elem2). A line is created tangent to elem1 and elem2. The location where the user selects the tangent arc or circle will be used to determine the desired line. See figure 6-6.



The left circle was selected at 1 and the right circle at 2 to generate line AB.

Figure 6-6: LINE TANGENT TO TWO CIRCLES

c. If PAR is selected, the user must select a line. An unbounded line is created tangent to elem1 and parallel to the selected line.

d. If PER is selected, the user must select any line, arc or circle (elem1). A line is created tangent to elem1 and perpendicular to elem2. The location where the user selects the perpendicular arc or circle will be used to determine the desired line.

e. If ANGLE is selected, the user must select a line and key in the number of degrees the angle forms with the selected line. A line is created tangent to elem1 at the entered angle from the selected line. All angles are measured in a counterclockwise direction.

4. PAR

The user must select a line (line1). The new line will be parallel to this line.

The following sub-menu will appear:

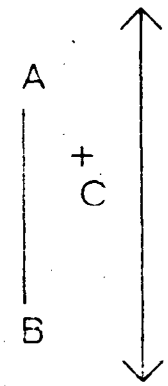
/POINT/TANG/PERP/OFF

a. If POINT is selected the user must specify a point. An unbounded line is created parallel to line1 through this point.

b. If TANG is selected, the user must select an arc or circle. An unbounded line is created parallel to line1 and tangent to the selected arc or circle. The location where the user selects the tangent arc or circle will be used to determine the desired line.

c. If PERP is selected, the user must select an arc or circle. An unbounded line is created parallel to line1 and perpendicular to the selected arc or circle. The location where the user selects the perpendicular arc or circle will be used to determine the desired line.

d. If OFF is selected, the user must key in an offset distance and must specify a direction point. An unbounded line is created parallel to line1 and offset the given distance. See figure 6-7.



Line AB and point C were selected and an offset was keyed in to create the unbounded line.

Figure 6-7: LINE CREATED AT AN OFFSET

5. PERP

The user must select a line, arc or circle. The new line will be perpendicular to the chosen element (elem1). The location where the user selects the perpendicular arc or circle will be used to determine the desired line.

The following sub-menu will appear:

/POINT/TANG/PAR/PERP/ANGLE/

a. If POINT is selected, the user must specify a point. A line is created perpendicular to elem1 through this point.

b. If TANG is selected the user must select an arc or circle (elem2). A line is created perpendicular to elem1 and tangent to elem2. The location where the user selects the tangent arc or circle will be used to determine the desired line.

c. If PAR is selected, the user must select a line (elem2). A line is created perpendicular to elem1 and parallel to elem2. This subcommand is invalid if elem1 is a line.

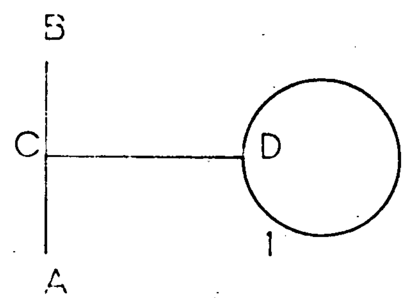
d. If PERP is selected, the user must select any line, arc or circle. (elem2). A line is created perpendicular to elem1 and elem2. This sub-command is invalid if both elem1 and elem2 are lines. The location where the user selects the perpendicular arc or circle will be used to determine the desired line. See figure 6-8. If ANGLE is selected, the user must select a line and key in the number of degrees the angle forms with the selected line. A line is created perpendicular to elem1 at the entered angle from the selected line. All angles are measured in a counterclockwise direction. This sub-command is invalid if elem1 is a line.

6. ANGLE

The user must select a line and key in the number of degrees the desired angle forms with this line. All angles are in a counterclockwise direction.

The following sub-menu will appear:

/POINT/TANG/PERP/BISECT/



Line AB was selected. The circle was then selected at point 1 to generate line CD.

Figure 6-8: PERPENDICULAR LINE

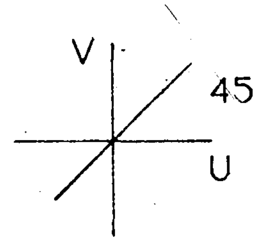
a. If POINT is selected, the user must specify a point. A line is created through the point at the desired angle from the selected line.

b. If TANG is selected, the user must select an arc or circle (elem1). A line is created tangent to elem1 at the desired angle fromt he selected line. The location where the user selects the tangent arc or circle will be used to determine the desired line.

c. If PERP is selected, the user must select an arc or circle (elem1). A line is created perpendicular to elem1 at the desired angle from the selected line.

The location where the user selects the perpendicular arc or circle will be used to determine the desired line.

d. If BISECT is selected, the user must select two intersecting lines. An unbounded line is created which bisects the given angle. See figure 6-9.



The U and V axes were keyed in to generate an unbounded line at 45 degrees.

Figure 6-9: LINE CREATED AT THE BISECTION OF AN ANGLE

6.2.3 CIRCLE Creation of a Circle Element

A circle can be created as an element in the working plane through the use of the CIRCLE function. The user can define circle parameters by specifying circumference points or a center point, by selecting tangent elements, or by keying in a radius. Where necessary, extensions of construction lines and arcs will be used (ie. Lines will be considered as unbounded lines, and arcs will be considered as circles).

When creating a circle the following main menu will appear:

CEN/CIR/

Each of the above sub-commands will listed in the above menu will be described.

1. CEN Center Point

   Any of the methods described in the POINT command "CP" can be used to define the center point.

2. CIR Circumference Point

   Any of the methods described in the POINT command "CP" can be used to define the circumference point.

3. Key in the Radius.

4. Select a Tangent Element.

   The user may also select any line, arc or circle which is tangent to the desired circle.

Subsequent menus will appear containing valid combinations of the CEN, CIR, Radius and Tangent Element options until the user has supplied sufficient information to define a circle. An error message will be produced if a circle cannot be drawn with the supplied parameters.

Upon creation of a circle, the user may reject the circle, select another circle sub-command, or return to the main menu in working plane mode.

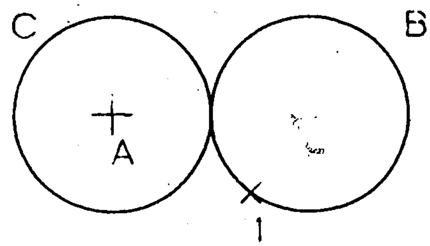
A circle can be created by:

1. Specifying a center point and keying in a radius.

2. Specifying a center point and a circumference point.

3. Specifying a center point and selecting a line, arc or circle which is tangent to the desired circle. The location where the user selects the tangent arc or circle will be used to determine the desired circle. See figure 6-10. Specifying two circumference points and keying in a radius. The user must also specify a reference point to determine which of the two possible circles is desired. See figure Specifying three circumference points.

4. Specifying two circumference points and selecting any line, arc or circle tangent to the desired circle. The location where the user selects the tangent arcs or circles will be used to determine which of the two possible circles is desired. See figure 6-12. Specifying one circumference point, keying in a radius, and selecting any element tangent to the desired circle. The location where the user selects the tangent element will be used to determine the desired circle. See figure

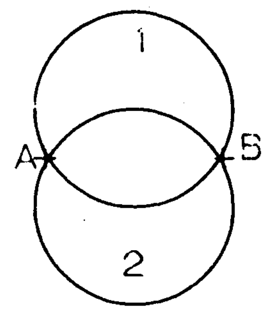


Circle B was selected at 1 to create circle C. Point A was selected as the center point.

Figure 6-10: CENTER POINT AND TANGENT

6-13. Selecting any two elements which are tangent to the desired circle and specifying a circumference point. A reference point must be specified to determine which circle is desired. Selecting any two elements which are tangent to the desired circle and keying in a radius. A reference point must be specified to determine which circle is desired. See figure 6-14.

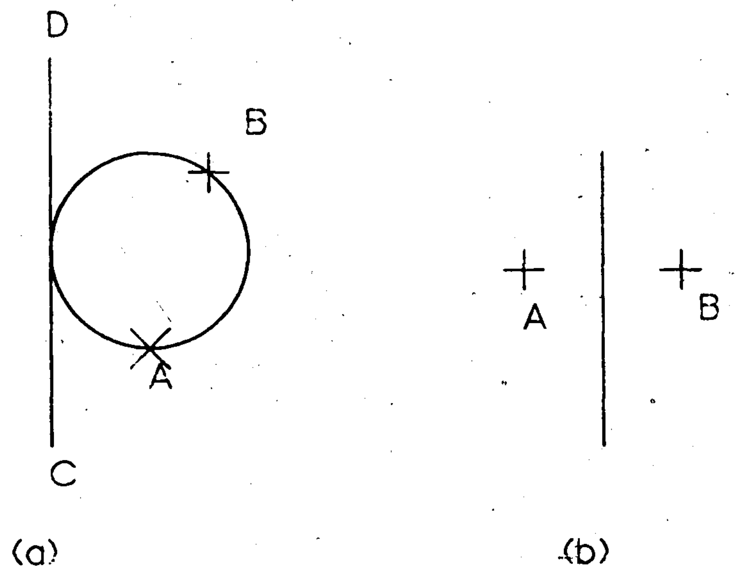
5. Selecting three intersecting lines which are tangent to the desired circle. A reference point must be provided to determine which circle is desired. See figure 6-15. Selecting two lines and one arc or circle which are tangent to the desired circle. A reference point must be provided to determine which circle is desired. See figure 6-16.



Points A and B were selected. The top circle was created by selecting 1 as the reference location and the bottom circle by selecting 2.

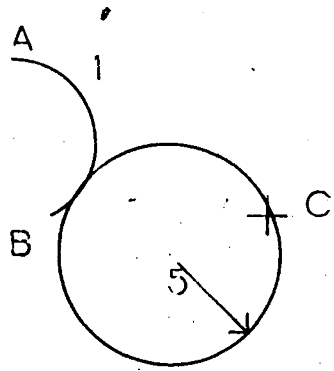
Figure 6-11: RADIUS AND TWO CIRCUMFERENCE POINTS

Selecting two arcs or circles and one line which are tangent to the desired circle. A reference point must be specified to determine which circle is desired. See figure 6-17.



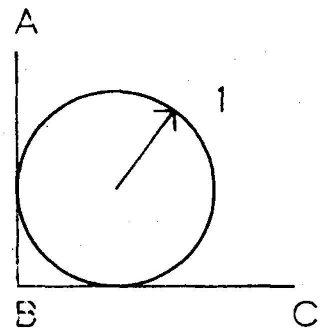
Points A and B and line CD were selected to create the circle in figure (a). Figure (b) is invalid.

Figure 6-12: TANGENT AND TWO CIRCUMFERENCE POINTS



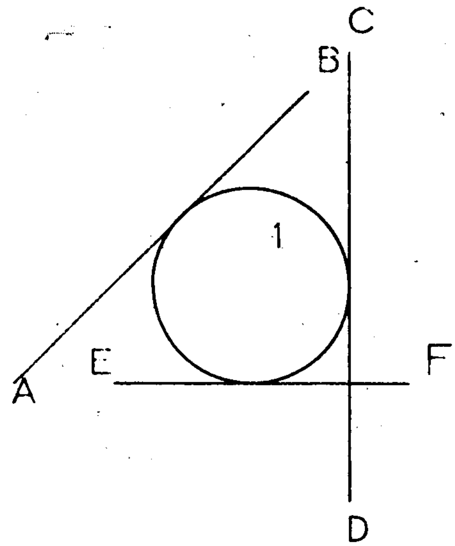
Arc AB was selected at 1, point C
was selected and a radius
of 5 was keyed in to create the circle.

Figure 6-13: RADIUS, TANGENT AND CIRCUMFERENCE POINT



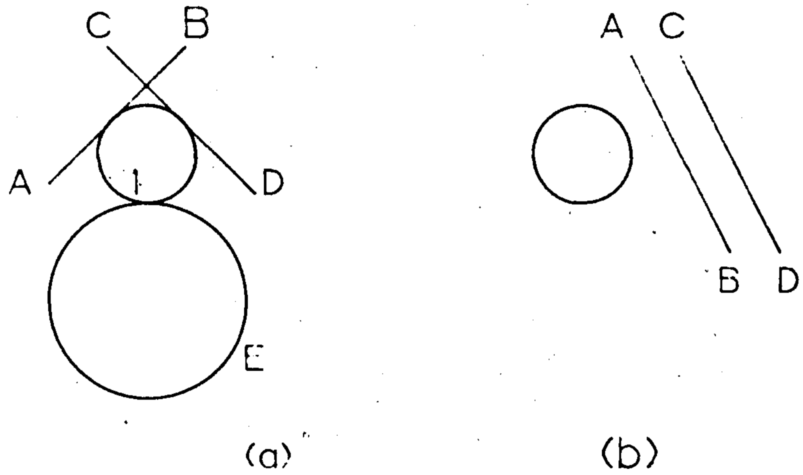
Lines AB and BC were selected.
A radius was keyed in and point 1 was used
This process is useful for creating fillets.

Figure 6-14: RADIUS AND TWO TANGENT LINES



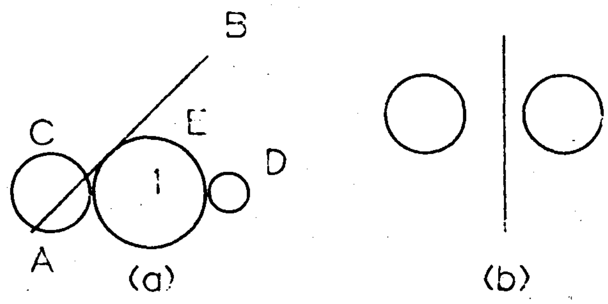
Lines AB, CD, and EF were selected.
Location 1 was used as a
reference to create the circle.

Figure 6-15: CIRCLE CREATED TANGENT TO THREE LINES



Circle E and lines AB and CD were selected.
Location 1 was used as the reference point to create the circle in figure (a).
Figure (b) is invalid.

Figure 6-16: TWO TANGENT LINES AND A CIRCLE



Line AB and circles C and D were selected.
Location 1 was used as the reference point to create circle E.
Figure (b) is invalid.

Figure 6-17: TWO TANGENT CIRCLES AND A LINE

6.2.4 CONLIN Creation of a Connected Linestring

A series of connected lines and arcs can be created in the working plane through the use of the CONLIN function. Each of the lines and arcs is considered an element in the working plane.

When using CONLIN, complex constructions aids are not available to create an arc or line. The user can construct points in advance however, to be used as vertices with the CONLIN command. The user is limited though, to specifying three points to define an arc (See ARC below).

The CONLIN function is used for input only to a working plane linestring. To delete any element in the linestring, the "ERS" command in the EDITEL function must be used.

The following sub-command can be accessed in CONLIN:

ARC Begin the Arc Sub-procedure

An arc is entered into a string of connected lines using the last point entered before the ARC command as the first end point, the next entered point as the center and the following point to determine the second end point. For example, in figure 6-18 the straight line segments 3-4 and 6-7 and the arc 4-6 can be entered with the sequence 3,4,arc,5,6,7. The distance between the first end point and the center point is the radius, so the second point to determine the next endpoint can be anywhere along a straight line through 5 and 6. Of the 2 possible arcs between the end points (clockwise and counterclockwise), the system chooses the one whose slope at the first end point is closest to the preceding line segment. A smooth arc will be drawn.

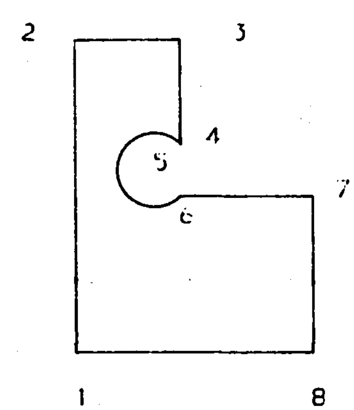


Figure 6-18: CREATION OF A CONNECTED LINESTRING

The REJ sub-command will reject the arc. The OPP sub-command will erase the arc and draw it in the opposite direction.

6.3 EDIT WORKING PLANE ELEMENTS

Elements in the working plane can be manipulated in many of the same ways as solids. Elements can be erased, moved, turned, scaled, reflected and copied. In addition, there are edit commands particular to 2D elements such as GRP, BRK, RELIM, and CLOSE. The user is restricted to editing elements only in the current working plane.

EDITEL Edit Elements

This command will activate the following main edit menu:

GRP/COPY//ERS/MOVE/TURN/SCL/REF/RELIM/BRK/CLOSE/F/NV/RET

A cancel command entered while editing an element will return the user to this menu.

Each of the commands listed in the above menu will be described.

1. GRP Group elements

   A group is a temporary collection of elements to be used with the ERS, MOVE, TURN, SCL and REF commands. When the CURGRP option is used in each of the above commands all elements in the current group are selected at one time.

   When the GRP command is entered, the user may:

   - START Start a new group

     A new group will be established. The user must select the elements to be included in the group. It should be noted that only one group can exist at a time. This group will remain active until either a new group is started or the user exits from the working plane.

- Modify a group

  The user can modify the group by selecting any element to add to, or delete from, the group. If the selected element currently resides in the group, it will be removed from the group. If the selected element is not in the group, it will be added to the group. This sub-function is the default for this command.

- HIGRP Highlight the group

  All elements in the group will be highlighted.

- HIELM Highlight the currently selected element

  The currently selected element will be highlighted.

2. COPY/NOCOPY Copy elements

   This command reverses the copy mode. If copy is selected, the copy mode is activated. Copy mode will remain active until the user selects NOCOPY or exits from the working plane.

   Copy mode is used in conjunction with the following commands: MOVE, TURN, SCL, and REF. If the copy mode is active, a replica of the elements to be edited remains in its original position. See figure 6-19.

3. ERS Erase elements

   The user must specify which elements are to be erased. This can be done by selecting the elements, using the current group (CURGRP), or by using the currently selected element (CURELM).

4. MOVE Move elements

Elements are moved an amount specified by a relative displacement. The user must enter two points (in 2D) to determine the displacement. The elements to be moved can be chosen by selecting the elements, using the current group (CURGRP), or by using the currently selected element (CURELM).

The OPP subcommand will redraw the specified elements with the opposite displacement. The REJ subcommand will undo the move and prompt for the next element(s).

5. TURN Turn elements

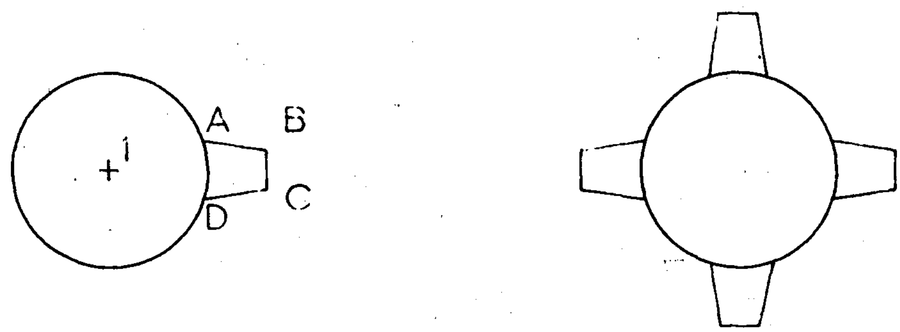
Elements are rotated about a stationary point on the working plane. The first point entered will remain stationary during rotation. The user can then enter the angle of rotation (in a counterclockwise direction), or enter two more points as the start and end points. The element(s) will be rotated the number of degrees required to rotate from the start point to the end point. The axis of rotation is through the stationary point and perpendicular to the working plane. See figure 6-19.

The elements to be rotated can be determined by selecting the elements, using the current group (CURGRP), or by using the currently selected element (CURELM).

The OPP subcommand will redraw the specfied elements with the opposite angle of rotation. The REJ subcommand will undo the rotation and ask for the next element(s) to be turned.

6. SCL Scale elements

Selected elements are scaled about a stationary point. The user must specify the stationary point and key in the scale factor. The size of the elements(s) will be changed by



(a) (b)

Lines AB, BC and CD were selected as a group under the GRP command. Point 1 was selected as the stationary point and an angle of 90 degrees was keyed in. The CURGRP was then chosen 3 times with copy mode on to create figure (b).

Figure 6-19: TURN A GROUP

the same amount in both the u and v directions.

The elements to be scaled can be determined by selecting the elements, by using the current group (CURGRP), or by using the currently selected element (CURELM).

The REJ subcommand will undo the scaling and prompt for the next elements(s).

7. REF Reflect elements about a line

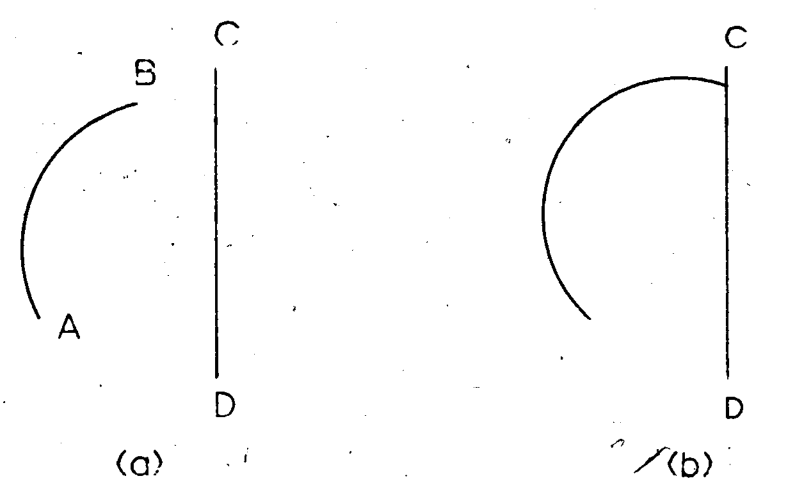
Selected elements are reflected about an arbitrary line. A mirror image of these elements is produced. The user may enter two points or select a line to determine the line about which the reflection will be done.

The user can specify the elements to be reflected by selecting the elements, using the current group (CURGRP), or by using the currently selected element (CURELM).

The REJ subcommand will undo the reflection and prompt the user for the next element(s).

8. RELIM Relimit an element

This command is used to change the length of a selected arc or line. The user must indicate which end of the element is to be modified and then select the limiting element. See figure 6-20.



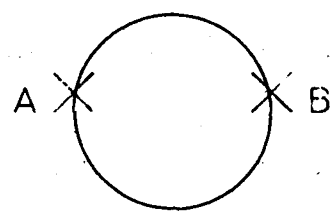
Arc AB was selected as the element to relimit. Line CD was selected as the intersecting element to generate figure (b).

Figure 6-20: RELIMIT AN ARC

The REJ subcommand will undo the relimit and prompt the user for the next element(s).

9. BRK Break an element

This command will break any selected line, arc, or circle into two elements. The user must indicate with the display device where the break should occur. Two break points must be indicated for a circle. See figure 6-21.



The circle was selected at point A. Point B was selected as an additional break point to create two arcs, AB and BC.

Figure 6-21: BREAK A CIRCLE

The REJ subcommand will undo the break and prompt for the next element(s).

10. CLOSE Close an element

This command converts a selected arc into a circle, or a selected line into an unlimited line.

The REJ subcommand will undo the close and prompt for the next element(s).

11. F N Refacet an arc or circle

This command is to be used only when an arc or circle in the working plane is to be translated or rotated to form a solid (using the PR or PT commands). The user must key in the number of facets N, and then select the arc or circle desired.

This command has no effect on the drawing of the arc or circle in the working plane.

12. NV Name vertex

This command permits names to be associated with or removed from vertices in the model. The user may specify any vertex in 3-space. The name can be shown on the graphic display or not. The user makes this decision at input time, but can subsequently change that decision using the NSHN (no show name) and SHN (show name) commands.

6.4 MODEL COORDINATE SYSTEMS

The reference coordinate system for the model is called the World Coordinate System (WCS). An axis indicator with 6 legs is displayed on the screen, indicating the orientation of the 3 mutually perpendicular axes (x,y,z) of the WCS and their negatives (See figure 6-22). The legs have the same lengths in 3-dimensional space, but as the view point is changed, the orientations and lengths of the vectors in the axis indicator also change to reflect the new view point.

A second coordinate system called the Local Coordinate System (LCS) (at any orientation or origin in 3-space) can be defined interactively. Its directions (u,v,w) are shown on a second axis indicator. Relative u, v or w coordinates can be indicated by using the respective axis as a subcommand. Absolute coordinates in the LCS can be indicated by using the LOC sub-function. Initially, the origin of the LCS is set at (0,0,0) in the WCS and the u, v and w axes are oriented in the x, y and z directions respectively.

LCS LOCAL COORDINATE SYSTEM

The user can manipulate the LCS through the use of the following sub-functions:

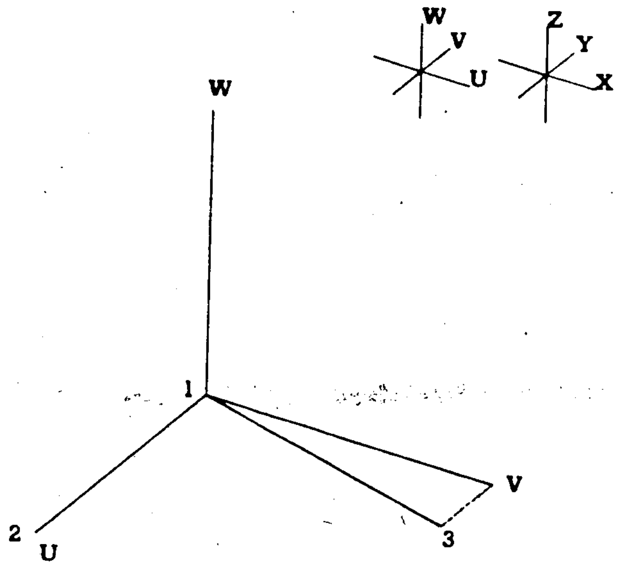


Figure 6-22: MODEL COORDINATE SYSTEMS

1. Indicate Three Points

The orientation of the LCS (with respect to its current origin) can be defined by indicating 3 points (the points 1, 2 and 3 of figure 6-22). Direction 12 is the +u axis of the LCS. The +v axis is perpendicular to 12 and is in the 123 plane. The +w axis is perpendicular to the u and v axes in a direction forming a right handed coordinate system.

This sub-function is the default for LCS.

2. A Rotate About an Axis

The orientation of the LCS (with respect to its current origin) can be defined by indicating any of the axes, u, v, w, x, y or z and the relative number of degrees of counter clockwise rotation desired.

3. TRAN Translate the Origin

The origin of the LCS is moved to a selected point. The orientation (direction cosines) of the u,v, and w axes remain unchanged.

4. SNAPWP Snap the LCS to a Selected Working Plane

The LCS is alligned to the origin and orientation of a selected working plane. The w axis is perpendicular to the working plane.

5. RES Reset to World Coordinate System

The LCS is reset to the World Coordinate System.

7. DATA STRUCTURES

7.1 MODEL FILE

The current IBM SOLID system employs a hierarchical data structure. That is, the objects or nodes in a model are related to one another by a series of pointers forming a tree structure. For example, in figure 7-1, node 1 has a pointer to its "dad" (the model node), to its "brother" (node 2), and to its first "son" (node 4).

In addition to these hierarchical pointers, the node contains the object's orientation in 3D space, the object type (cuboid, cylinder, non-primitive etc.), and a pointer to a polyhedron list. The polyhedron list contains information on the surfaces, edges, and vertices of the object.

The working plane is a node in the model hierarchy. It is considered a primitive or the lowest level of object (like cuboid or cylinder) thus; it has no nodes below it in the tree.

Unlike a solid, the polyhedron list for the working plane contains only edge and vertex information. In addition, the IBM SOLID system represents all objects by using line edges only. The concept of arc and point edges was introduced in the working plane. A summary of the polyhedron list for a working plane is shown in table

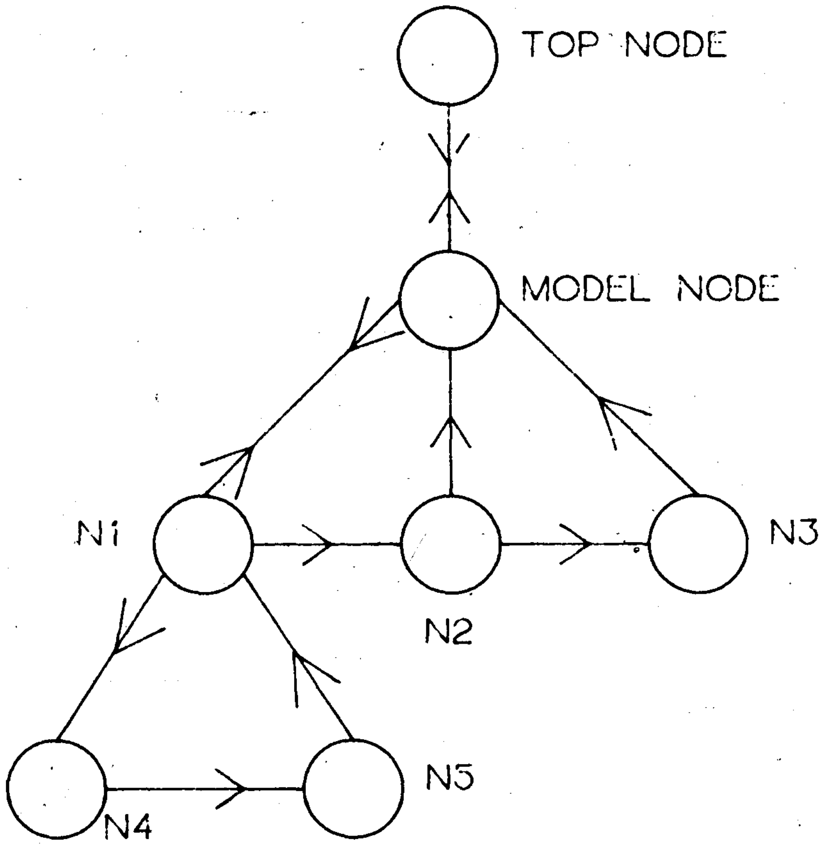


Figure 7-1: IBM SOLID HIERARCHICAL DATA STRUCTURE

7-1. Note that mathematical representations of arcs and circles are used rather than straight line approximations.

The bitword in table 7-1 is defined as follows:

- Bits 1 and 2 are used by solids only.
- Bit 3 signifies an arc or circle.

Table 7-1: WORKING PLANE POLYHEDRON LIST

ELEMENT TYPE	BITWORD	FIELD1	FIELD2	FIELD3
Arc	00100c	Start	End	Center
Arc 180	00101c	Start	End	Circum
Arc 360	00100c	Start	Circum	Center
Point	000000	Point		
Line LL	000100	Start	End	-
Line UU	000111	Start	End	-
Line UL	000110			
Line LU	000101	Start	End	-

- Bit 4 signifies a line.
- Bits 5 and 6 have various meanings. If bit 4 is set, then bits 5 and 6 indicate unlimited portions of a line. If bit 3 is set, then bits 5 and 6 determine the type of arc or circle.
- Bit 7 is a counterclockwise bit for arcs.

7.2 DISPLAY FILE

The IBM SOLID system uses a display list when drawing objects on the screen. This display list contains only those objects which are at least partially visible on the screen. The graphics routines then clip the objects in the display list and draw them on the screen. The use of a display list minimizes the frequency of display regeneration. Thus, if additional elements are to be added to the display, no regeneration is necessary.

Since all objects in the IBM SOLID system are represented by straight lines only, the display list had to be modified to handle the working plane elements of points, arcs and circles. The format for the working plane display list is shown in figure 7-2.

The display list design not only enables efficient drawing of working plane elements, but also efficient element selection as well. For example, if the user is searching for a line in the working plane, the display list for a line need only be searched.

The following steps are used to generate and to draw the display list.

- First, a box test is conducted on the entire working plane to determine whether it lies at least partially on the screen.

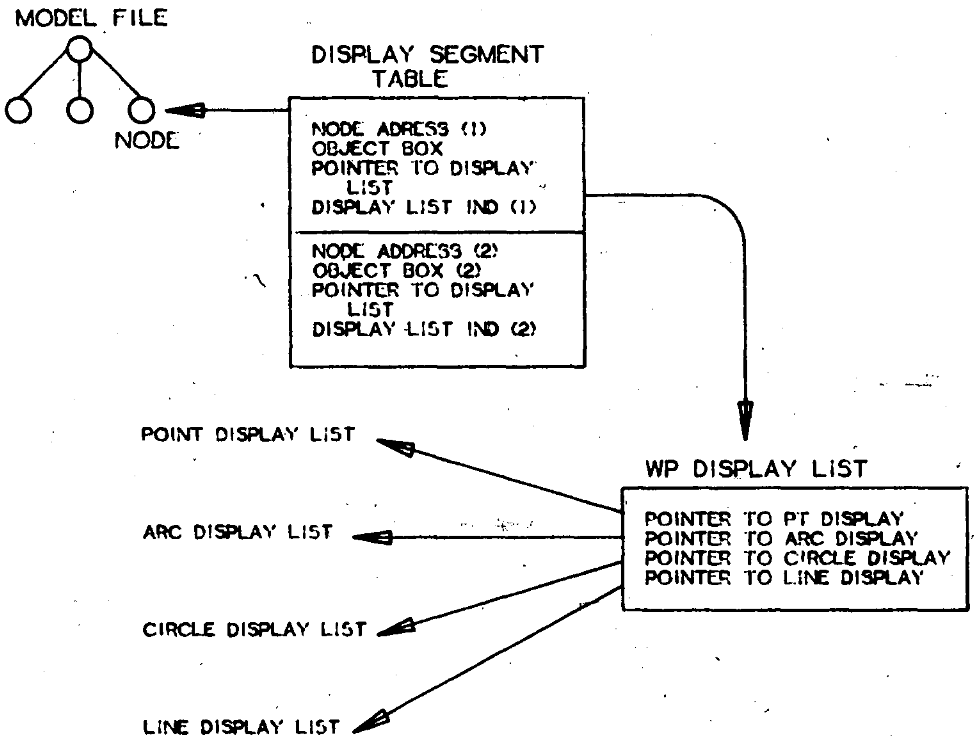


Figure 7-2: WORKING PLANE DISPLAY LIST

- If the working plane lies on the screen, a box test is then conducted on each element in the working plane to determine whether it lies at least partially on the screen.

- If the element lies on the screen, its parameters are projected to the screen and entered into the display list. Thus, circles in the model are stored as ellipses in the display list to handle various viewing angles.

- The draw routine then generates line segments for arcs, circles and points "on the fly". This greatly reduces the size of the display list for arcs and circles. An efficient line generation routine by Smith[11] will be used to generate the line segments for arcs and circles.

- The graphics routine then clips each of the elements in the display list to the screen and then draws them.

---

11 L. Smith, "Drawing Ellipses, Hyperbolas or Parabolas With a Fixed Number of Points and Maximun Inscribed Area," Computer Journal, Vol. 14, 1969, p.81.

8. CONCLUSIONS

The charge of this thesis was to design a system to enhance IBM SOLID to include geometric constructions in a plane. The proposed system provides the capability for constructing and editing 2D elements in a defined working plane. From input recieved from the pilot location using IBM SOLID, this capability is a much needed one.

The benefits of planar constructions are many and include:

1. Increased Productivity. Many hours of mathematical calculations should be reduced drastically. Designers need not calculate exact coordinates but only select elements and specify conditions (ie. tangent, parallel etc.).

2. Reduced Skill Level of Operator. The skill level of operators can be reduced as mathematical expertise will not be required. Thus the system can be used by mechanical designers and draftsmen alike.

3. Reduced Errors. The probability of design errors will be reduced since many mathematical calculations will be performed automatically.

4. Increased Flexibility. The user can work in any dimension that is easiest or most comfortable for him. A user may create a 2D secton of a complicated part and sweep it into a solid, rather than to build the part from solid primitives.

5. Reduced Complexity. A complex drawing can be reduced to planar views. For example, a user could establish a local coordinate system and work in a plane to simplify the input of coordinates and to deal with only two dimensions at a time.

6. Increased Acceptance. This system bridges the gap between 2D and 3D worlds. Users that are accustomed to working in a 2D environment may accept the transfer to solids more readily if they are introduced to familiar concepts.

9. AREAS OF FUTURE STUDY

The addition of planar construction aids was the first phase of enhancements to IBM SOLID. Subsequent modifications include the entry and display of dimensions and tolerances in 3D space. These would be stored in the 3D model and would be associated appropriately with the solid object. Dimensioning presents some unique problems when applied to 3D space. For example, problems arise when viewing angles make the dimensions unreadable.

The addition of tolerances would open "Pandora's Box" to a myriad of applications. Tolerances, like dimensions, could be used for design documentation. In addition, tolerances could be used with interference checking to simulate assembly "fits". Worst case tolerance buildups could be analyzed.

10. BIBLIOGRAPHY

1. Adams, J. and Rogers, D. MATHEMATICAL ELEMENTS FOR COMPUTER GRAPHICS. New York: McGraw Hill, 1976.

2. Chasen, S. GEOMETRIC PRINCIPLES AND PROCEDURES FOR COMPUTER GRAPHIC APPLICATIONS. Englewood Cliffs, New Jersey: Prentice-Hall, Inc., 1978.

3. Fitzgerald, W., Gracer, F., Wolfe R. "GRIN: Interactive Graphics for Modeling Solids," IBM JOURNAL OF RESEARCH DEVELOPMENT, 25, 1981, pp. 281-294.

4. Hillyard, R. "The Build Group of Solid Modelers," IEEE GRAPHICS, March, 1982, p. 43.

5. Kinnucan, P. "Solid Modelers Make the Scene," HIGH TECHNOLOGY, July/August, 1982, p.38.

6. Newman, W. and Sproull, R. PRINCIPLES OF INTERACTIVE COMPUTER GRAPHICS. New York: McGraw-Hill, 1979.

7. Requicha, A., Voelcker H., "Solid Modeling: A Historical Summary and Contemporary Assessment," IEEE GRAPHICS, March, 1982, p. 9.

8. Smith, L.B. "Drawing Ellipses, Hyperbolas or Parabolas With a Fixed Number of Points and Maximum Inscribed Area," COMPUTER JOURNAL, Vol. 14, 1969, p. 81.

VITA

Ms. Carol Riggin was born on November 5, 1956 in Pompton Plains, New Jersey to Mr. and Mrs. George Richardi.

She graduated from DePaul High School in June, 1974 as salutatorian. She graduated magna cum laude from Lehigh University in June, 1978 with a Bachelor of Science in Industrial Engineering.

Ms. Riggin was then employed by A.T.T. Long Lines in Oakton, Virginia where she held a variety of managerial positions. She returned to Lehigh University in January, 1982 for a Master of Science in Industrial Engineering. While a graduate student, Ms. Riggin recieved a both a teaching and research assistantship in the area of CAD/CAM. Her scholastic emphasis was in information systems.